Foundation of Cryptography, Lecture 4
Pseudorandom Functions

Iftach Haitner, Tel Aviv University

Tel Aviv University.

April 23, 2013

Section 1

Function Families

function families

1. $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$, where $\mathcal{F}_n = \{f : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
2. We write $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$
3. If $m(n) = \ell(n) = n$, we omit it from the notation
4. We identify function with their description
5. The rv F_n is uniformly distributed over $\mathcal{F}_n$

Efficient function families

Definition 1 (efficient function family)

An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n\in\mathbb{N}}$ is efficient, if:

Samplable. $\mathcal{F}$ is samplable in polynomial-time: there exists a PPT that given 1^n, outputs (the description of) a uniform element in $\mathcal{F}_n$.

Efficient. There exists a polynomial-time algorithm that given $x \in \{0,1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

Random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, let $\Pi_{m,\ell}$ be the family of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

Random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, let $\Pi_{m,\ell}$ be the family of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $2^m \cdot \ell$ bits to describe an element (i.e., function) of $\Pi_{m,\ell}$.

Random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, let $\Pi_{m,\ell}$ be the family of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $2^m \cdot \ell$ bits to describe an element (i.e., function) of $\Pi_{m,\ell}$.
- We sometimes think of $\pi \in \Pi_{m,\ell}$ as a random string of length $2^m \cdot \ell$.

Random functions

Definition 2 (random functions)

For $m, \ell \in \mathbb{N}$, let $\Pi_{m,\ell}$ be the family of all functions from $\{0,1\}^m$ to $\{0,1\}^\ell$.

- It takes $2^m \cdot \ell$ bits to describe an element (i.e., function) of $\Pi_{m,\ell}$.
- We sometimes think of $\pi \in \Pi_{m,\ell}$ as a random string of length $2^m \cdot \ell$.
- Let $\Pi_n = \Pi_{n,n}$

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{m(n),\ell(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{m(n),\ell(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $\ell(n) = n$ (why?)

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left| \Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{m(n),\ell(n)}}(1^n) = 1 \right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $\ell(n) = n$ (why?)
2. Easy to construct (with no assumption!) for $m(n) = \log n$ and $\ell \in \mathrm{poly}$

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{m(n),\ell(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $\ell(n) = n$ (why?)
2. Easy to construct (with no assumption!) for $m(n) = \log n$ and $\ell \in \operatorname{poly}$
3. PRF imply a PRG

Pseudorandom functions

Definition 3 (pseudorandom functions)

A function family ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^{m(n)} \mapsto \{0,1\}^{\ell(n)}\}$ is pseudorandom, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_{m(n),\ell(n)}}(1^n) = 1\right| = \operatorname{neg}(n),$$

for any oracle-aided PPT D.

1. Suffices to consider $\ell(n) = n$ (why?)
2. Easy to construct (with no assumption!) for $m(n) = \log n$ and $\ell \in \text{poly}$
3. PRF imply a PRG
4. Pseudorandom permutations (PRPs)

Section 2

PRF from OWF

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s, x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s \colon s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s,x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s \colon s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

- Alternative definition, $f_s(x) := f_{g_{x_1}(s)}(x_{2,\ldots,|x|})$ (letting $f_s(\lambda) = s$)

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s, x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s : s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

- Alternative definition, $f_s(x) := f_{g_{x_1}(s)}(x_{2,\ldots,|x|})$ (letting $f_s(\lambda) = s$)
- g is poly-time computable $\implies$ $\mathcal{F}$ is efficient.

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s, x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s : s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

- Alternative definition, $f_s(x) := f_{g_{x_1}(s)}(x_{2,\ldots,|x|})$ (letting $f_s(\lambda) = s$)
- g is poly-time computable $\implies$ $\mathcal{F}$ is efficient.

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s, x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s : s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

- Alternative definition, $f_s(x) := f_{g_{x_1}(s)}(x_{2,\ldots,|x|})$ (letting $f_s(\lambda) = s$)
- g is poly-time computable $\implies$ $\mathcal{F}$ is efficient.

Theorem 5 (Goldreich-Goldwasser-Micali (GGM))

If g is a PRG then $\mathcal{F}$ (defined above) is a PRF.

the construction

Construction 4

For $g\colon \{0,1\}^n \mapsto \{0,1\}^{2n}$, let $g_0(s) = g(s)_{1,\ldots,n}$ and $g_1(s) = g(s)_{n+1,\ldots,2n}$. For $s, x \in \{0,1\}^*$ define f_s as

$$f_s(x) = g_{x_n}(\ldots(g_{x_2}(g_{x_1}(s))))$$

Let $\mathcal{F}_n = \{f_s\colon s \in \{0,1\}^n\}$ and $\mathcal{F} = \{\mathcal{F}_n\}$.

- Alternative definition, $f_s(x) := f_{g_{x_1}(s)}(x_{2,\ldots,|x|})$ (letting $f_s(\lambda) = s$)
- g is poly-time computable $\implies$ $\mathcal{F}$ is efficient.

Theorem 5 (Goldreich-Goldwasser-Micali (GGM))

If g is a PRG then $\mathcal{F}$ (defined above) is a PRF.

Corollary 6

OWFs imply PRFs.

Proof Idea

- Easy to prove for inputs of length 2.

Proof Idea

- Easy to prove for inputs of length 2.

  Observation: $\mathsf{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

Proof Idea

- Easy to prove for inputs of length 2.

  Observation: $\mathsf{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

  Proof: $\mathsf{D}' = (g(U_n^{(0)}), g(U_n^{(1)})) \approx_c U_{4n}$ and $\mathsf{D} \approx_c \mathsf{D}'$.

Proof Idea

- Easy to prove for inputs of length 2.

  Observation: $\mathsf{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

  Proof: $\mathsf{D}' = (g(U_n^{(0)}), g(U_n^{(1)})) \approx_c U_{4n}$ and $\mathsf{D} \approx_c \mathsf{D}'$.
- Hence we can handle input of length 2

Proof Idea

- Easy to prove for inputs of length 2.

  Observation: $\mathsf{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

  Proof: $\mathsf{D}' = (g(U_n^{(0)}), g(U_n^{(1)})) \approx_c U_{4n}$ and $\mathsf{D} \approx_c \mathsf{D}'$.
- Hence we can handle input of length 2
- Extend to longer inputs?

Proof Idea

- Easy to prove for inputs of length 2.

  Observation: $\mathsf{D} = (g(g_0(U_n)), g(g_1(U_n)))$ is pseudorandom:

  Proof: $\mathsf{D}' = (g(U_n^{(0)}), g(U_n^{(1)})) \approx_c U_{4n}$ and $\mathsf{D} \approx_c \mathsf{D}'$.
- Hence we can handle input of length 2
- Extend to longer inputs?
- We show that an efficient sample from the *truth table* of $f \leftarrow \mathcal{F}_n$, is computationally indistinguishable from that of $\pi \leftarrow \Pi_n$.

The Actual Proof

Assume $\exists$ PPT D, $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{F_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$, and fix $n \in \mathbb{N}$

The Actual Proof

Assume $\exists$ PPT D, $p \in$ poly and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$\left|\Pr[\mathsf{D}^{F_n}(1^n) = 1] - \Pr[\mathsf{D}^{\Pi_n}(1^n) = 1]\right| \geq \frac{1}{p(n)}, \tag{1}$$

for any $n \in \mathcal{I}$, and fix $n \in \mathbb{N}$

Let $t = t(n) \in$ poly be a bound on the running time of $\mathsf{D}(1^n)$. We use D to construct a PPT D′ such that

$$\left|\Pr[\mathsf{D}'(U_{2n}^t) = 1] - \Pr[\mathsf{D}'(g(U_n)^t) = 1\right| > \frac{1}{np(n)},$$

where $U_{2n}^t = U_{2n}^{(1)}, \ldots, U_{2n}^{(t(n))}$ and $g(U_n)^t = g(U_n^{(1)}), \ldots, g(U_n^{(t(n))})$.

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$?

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$?

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$? We emulate it from D's point of view.

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$? We emulate it from D's point of view.
- We present efficient “function family" $\mathcal{O}_k = \{O_k^{s^1,\ldots,s^t}\}$ s.t.

for any $k \in [n]$, where H_K is uniformly sampled from $\mathcal{H}_k$.

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0,\ldots,n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$? We emulate it from D's point of view.
- We present efficient “function family" $\mathcal{O}_k = \{O_k^{s^1,\ldots,s^t}\}$ s.t.
  - $\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

for any $k \in [n]$, where H_K is uniformly sampled from $\mathcal{H}_k$.

The Hybrid

Let g and f be as in the definition of $\mathcal{F}_n$

Definition 7

For $k \in \{0, \ldots, n\}$, let $\mathcal{H}_k = \{h_\pi \colon \{0,1\}^n \mapsto \{0,1\}^n \colon \pi \in \Pi_{k,n}\}$, where

- $h_\pi(x) = f_{\pi(x_{1,\ldots,k})}(x_{k+1,\ldots,n})$
- $f_y(\lambda) = y$ (Hence, $\mathcal{H}_n = \Pi_n$)
- $\Pi_{0,n} = \{0,1\}^n$, and for $\pi \in \Pi_{0,n}$ let $\pi(\lambda) = \pi$ (Hence, $\mathcal{H}_0 = \mathcal{F}_n$)

- Note that $\mathcal{H}_0 = \mathcal{F}_n$ and $\mathcal{H}_n = \Pi_n$
- Can we emulate $\mathcal{H}_k$? We emulate it from D's point of view.
- We present efficient "function family" $\mathcal{O}_k = \{O_k^{s^1,\ldots,s^t}\}$ s.t.
  - $\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$
  - $\mathsf{D}^{O_k^{g(U_n)^t}}(1^n) \equiv \mathsf{D}^{H_{k-1}}(1^n)$

  for any $k \in [n]$, where H_K is uniformly sampled from $\mathcal{H}_k$.

Completing the Proof

Let $\mathsf{D}'(y)$ return $\mathsf{D}^{O_k^y}(1^n)$ for k uniformly chosen in $[n]$.

Completing the Proof

Let $\mathrm{D}'(y)$ return $\mathrm{D}^{O_k^y}(1^n)$ for k uniformly chosen in $[n]$. Hence

$$
\begin{aligned}
&\left|\Pr[\mathrm{D}'(U_{2n}^t) = 1]\right| - \Pr[\mathrm{D}'(g(U_n)^t) = 1] \\
= &\left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k^{U_{2n}^t}}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathrm{D}^{O_k^{g(U_n)^t}}(1^n) = 1]\right|
\end{aligned}
$$

Completing the Proof

Let $\mathsf{D}'(y)$ return $\mathsf{D}^{\mathcal{O}_k^y}(1^n)$ for k uniformly chosen in $[n]$. Hence

$$\begin{aligned}
&\left|\Pr[\mathsf{D}'(U_{2n}^t) = 1]\right| - \Pr[\mathsf{D}'(g(U_n)^t) = 1] \\
&= \left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathsf{D}^{\mathcal{O}_k^{U_{2n}^t}}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[\mathsf{D}^{\mathcal{O}_k^{g(U_n)^t}}(1^n) = 1]\right| \\
&= \frac{1}{n}\left|\sum_{k=1}^{n} \Pr[\mathsf{D}^{H_k}(1^n) = 1] - \sum_{k=1}^{n} \Pr[\mathsf{D}^{H_{k-1}}(1^n) = 1]\right|
\end{aligned}$$

Completing the Proof

Let $D'(y)$ return $D^{O_k^y}(1^n)$ for k uniformly chosen in $[n]$. Hence

$$\begin{aligned}
&\left|\Pr[D'(U_{2n}^t) = 1]\right| - \Pr[D'(g(U_n)^t) = 1] \\
&= \left|\sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[D^{O_k^{U_{2n}^t}}(1^n) = 1] - \sum_{k=1}^{n} \frac{1}{n} \cdot \Pr[D^{O_k^{g(U_n)^t}}(1^n) = 1]\right| \\
&= \frac{1}{n}\left|\sum_{k=1}^{n} \Pr[D^{H_k}(1^n) = 1] - \sum_{k=1}^{n} \Pr[D^{H_{k-1}}(1^n) = 1]\right| \\
&= \frac{1}{n}\left|\Pr[D^{H_n}(1^n) = 1] - \Pr[D^{H_0}(1^n) = 1]\right| = \frac{1}{np(n)} \square
\end{aligned}$$

The family $\mathcal{O}_k$

$\mathcal{O}_k := \{O_k^{s^1,\ldots,s^t} : s^1,\ldots,s^t \in \{0,1\}^n \times \{0,1\}^n\}$.

Algorithm 8 ($O_k^{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x^i_k}$ (where ℓ is the minimal such index).
   Otherwise, set $z = s^i_{x^i_k}$ (for $k = 0$ set $z = s^1_0$).
2. Return $f_z(x^i_{k+1,\ldots,n})$

The family $\mathcal{O}_k$

$\mathcal{O}_k := \{O_k^{s^1,\ldots,s^t} : s^1,\ldots,s^t \in \{0,1\}^n \times \{0,1\}^n\}$.

Algorithm 8 ($O_k^{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x^i_k}$ (where ℓ is the minimal such index).
   Otherwise, set $z = s^i_{x^i_k}$ (for $k = 0$ set $z = s^1_0$).
2. Return $f_z(x^i_{k+1,\ldots,n})$

- $\mathcal{O}_k$ is stateful.

The family $\mathcal{O}_k$

$\mathcal{O}_k := \{O_k^{s^1,\ldots,s^t} : s^1,\ldots,s^t \in \{0,1\}^n \times \{0,1\}^n\}$.

Algorithm 8 ($O_k^{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x^i_k}$ (where ℓ is the minimal such index). Otherwise, set $z = s^i_{x^i_k}$ (for $k = 0$ set $z = s^1_0$).
2. Return $f_z(x^i_{k+1,\ldots,n})$

- $\mathcal{O}_k$ is stateful.

The family $\mathcal{O}_k$

$\mathcal{O}_k := \{O_k^{s^1,\ldots,s^t} : s^1,\ldots,s^t \in \{0,1\}^n \times \{0,1\}^n\}$.

Algorithm 8 ($O_k^{s^1,\ldots,s^t}$)

On the i'th query $x^i \in \{0,1\}^n$:

1. If x^ℓ with $x^\ell_{1,\ldots,k-1} = x^i_{1,\ldots,k-1}$ was previously asked, set $z = s^\ell_{x^i_k}$ (where ℓ is the minimal such index). Otherwise, set $z = s^i_{x^i_k}$ (for $k = 0$ set $z = s^1_0$).
2. Return $f_z(x^i_{k+1,\ldots,n})$

- $\mathcal{O}_k$ is stateful.

Claim 9

$D^{O_k^{g(U_n)^t}}(1^n) \equiv D^{O_{k-1}^{U_{2n}^t}}(1^n)$ for all $k \in \{1,\ldots,n\}$.

$\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

$\mathsf{D}^{\mathcal{O}_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof?

$$\mathsf{D}^{\mathcal{O}_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?

$\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?

Let $\widetilde{O}_k$ be the variant of O_k that returns z (and not $f_z(x_{k+1,\ldots,n})$ as in Algorithm 8) and let $\widetilde{\mathsf{D}}_k$ be the algorithm that implements D using $\widetilde{O}_k$ (by computing $f_z(x_{k+1,\ldots,n})$ by itself).

$\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \mathsf{D}^{H_k}(1^n)$

Proposition 10

For any $\ell, m \in \mathbb{N}$ and any algorithm A, it holds that $A^{\Pi_{\ell,m}} \equiv A^{B_{\ell,m}}$, where the stateful random algorithm $B_{\ell,m}$ answers identical queries with the same answer, and answers new queries with a random string of length m.

Proof? Does the above trivialize the whole issue of PRF?

Let $\widetilde{O}_k$ be the variant of O_k that returns z (and not $f_z(x_{k+1,\ldots,n})$ as in Algorithm 8) and let $\widetilde{\mathsf{D}}_k$ be the algorithm that implements D using $\widetilde{O}_k$ (by computing $f_z(x_{k+1,\ldots,n})$ by itself).

By Proposition 10

$$\mathsf{D}^{O_k^{U_{2n}^t}}(1^n) \equiv \widetilde{\mathsf{D}}_k^{\widetilde{O}_k^{U_{2n}^t}}(1^n) \equiv \widetilde{\mathsf{D}}_k^{\pi_{k,n}}(1^n) \equiv \mathsf{D}^{H_k}(1^n) \tag{2}$$

$$D^{O_k^{g(U_n)^t}}(1^n) \equiv D^{H_{k-1}}(1^n)$$

$$\mathsf{D}^{O_k^{g(U_n)^t}}(1^n) \equiv \mathsf{D}^{H_{k-1}}(1^n)$$

Immediately follows by Claim 9 and Eq 2.

Section 3

PRP from PRF

Pseudorandom permutations

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 11 (pseudorandom permutations)

A permutation ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left|\Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1\right| = \operatorname{neg}(n), \tag{3}$$

for any oracle-aided PPT D

Pseudorandom permutations

Let $\widetilde{\Pi}_n$ be the set of all permutations over $\{0,1\}^n$.

Definition 11 (pseudorandom permutations)

A permutation ensemble $\mathcal{F} = \{\mathcal{F}_n : \{0,1\}^n \mapsto \{0,1\}^n\}$ is a pseudorandom permutation, if

$$\left| \Pr[\mathsf{D}^{\mathcal{F}_n}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_n}(1^n) = 1 \right| = \operatorname{neg}(n), \tag{3}$$

for any oracle-aided PPT D

- Eq 3 holds for any PRF

PRF to PRP

Definition 12 (LR)

Given $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, the permutation $\mathrm{LR}(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ is defined by

$$\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell).$$

Let $\mathrm{LR}^i(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be the i'th iteration of the above operation.

PRF to PRP

Definition 12 (LR)

Given $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, the permutation $\mathsf{LR}(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ is defined by

$$\mathsf{LR}(f)(\ell, r) = (r, f(r) \oplus \ell).$$

Let $\mathsf{LR}^i(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be the i'th iteration of the above operation.

Construction 13

Given a function family $\mathcal{F} = \{\mathcal{F}_n\colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let
$\mathsf{LR}^i(\mathcal{F}) = \{\mathsf{LR}^i(\mathcal{F}_n) = \{\mathsf{LR}^i(f)\colon f \in \mathcal{F}_n\}\}$,

PRF to PRP

Definition 12 (LR)

Given $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, the permutation $\mathrm{LR}(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ is defined by

$$\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell).$$

Let $\mathrm{LR}^i(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be the i'th iteration of the above operation.

Construction 13

Given a function family $\mathcal{F} = \{\mathcal{F}_n\colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i(\mathcal{F}_n) = \{\mathrm{LR}^i(f)\colon f \in \mathcal{F}_n\}\}$,

$\mathrm{LR}(\mathcal{F})$ is always a permutation family, and is efficient if $\mathcal{F}$ is.

PRF to PRP

Definition 12 (LR)

Given $f\colon \{0,1\}^n \mapsto \{0,1\}^n$, the permutation $\mathrm{LR}(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ is defined by

$$\mathrm{LR}(f)(\ell, r) = (r, f(r) \oplus \ell).$$

Let $\mathrm{LR}^i(f)\colon \{0,1\}^{2n} \mapsto \{0,1\}^{2n}$ be the i'th iteration of the above operation.

Construction 13

Given a function family $\mathcal{F} = \{\mathcal{F}_n \colon \{0,1\}^n \mapsto \{0,1\}^n\}$, let $\mathrm{LR}^i(\mathcal{F}) = \{\mathrm{LR}^i(\mathcal{F}_n) = \{\mathrm{LR}^i(f)\colon f \in \mathcal{F}_n\}\}$,

$\mathrm{LR}(\mathcal{F})$ is always a permutation family, and is efficient if $\mathcal{F}$ is.

Theorem 14 (Luby-Rackoff)

Assuming that $\mathcal{F}$ is a PRF, then $\mathrm{LR}^3(\mathcal{F})$ is a PRP

Proving Thm 14

Proving Thm 14

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 15

$|\Pr[\mathsf{D}^{\mathsf{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \leq \frac{4 \cdot q^2}{2^n}$,
for any q-query algorithm D.

Proving Thm 14

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 15

$|\Pr[\mathsf{D}^{\mathsf{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \leq \frac{4 \cdot q^2}{2^n}$,
for any q-query algorithm D.

- How would you prove Claim 15?

Proving Thm 14

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 15

$|\Pr[\mathsf{D}^{\mathsf{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \leq \frac{4 \cdot q^2}{2^n}$,
for any q-query algorithm D.

- How would you prove Claim 15?
- Start with non-adaptive D, and show that things only "get wrong" if $\mathsf{LR}(\ell, r) = \mathsf{LR}(\ell', r')$ for two different queries $(\ell, r) \neq (\ell', r')$

Proving Thm 14

It suffices to prove the the following holds for any $n \in \mathbb{N}$ (why?)

Claim 15

$|\Pr[\mathsf{D}^{\mathsf{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[\mathsf{D}^{\widetilde{\Pi}_{2n}}(1^n)| = 1] \leq \frac{4 \cdot q^2}{2^n}$,
for any q-query algorithm D.

- How would you prove Claim 15?
- Start with non-adaptive D, and show that things only "get wrong" if $\mathsf{LR}(\ell, r) = \mathsf{LR}(\ell', r')$ for two different queries $(\ell, r) \neq (\ell', r')$
- Can you bound the above probability?

Section 4

Applications

General paradigm

Design a scheme assuming that you have random functions, and the realize them using PRFs.

Private-key Encryption

Construction 16 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D})$:

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

Private-key Encryption

Construction 16 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D})$:

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?

Private-key Encryption

Construction 16 (PRF-based encryption)

Given an (efficient) PRF $\mathcal{F}$, define the encryption scheme $(\mathsf{Gen}, \mathsf{E}, \mathsf{D})$:

Key generation: $\mathsf{Gen}(1^n)$ returns $k \leftarrow \mathcal{F}_n$

Encryption: $\mathsf{E}_k(m)$ returns $U_n, k(U_n) \oplus m$

Decryption: $\mathsf{D}_k(c = (c_1, c_n))$ returns $k(c_1) \oplus c_2$

- Advantages over the PRG based scheme?
- Proof of security?